

04-10-00

A

04/07/00
jc772 U.S. PTO

IBM Docket No. RAL9000022

In the United States Patent and Trademark Office Patent Application Transmittal

Transmitted herewith for filing is the Patent Application of:

Inventors(s): Richard John Blasiak, David Wayne Cosby, Anthony Matteo Gallo, Frank Andrew Smoak

For: Method and System for Providing Remote Procedure Calls in a Multiprocessing System

jc542 U.S. PTO
09/545689

04/07/00

Enclosed are

- 11 pages of specification, including 10 claims, plus 6 sheets of drawings.
- X An assignment of the invention to International Business Machines Corporation, Armonk, New York 10504.
- A certified copy of a/an application.
- X Declaration and Power of Attorney.
- X PTO-1449 & references
- X A return post card
- Other:

Filing Fee Calculation (For Other Than Small Entity)

Basic Fee:						\$690.00
Claims Fees:	Filed	Limit	Extra		Rate per Extra	
Total claims:	10	20	0		\$18.00	\$0.00
Independent claims:	2	3	0		\$78.00	\$0.00
Multiple Dependent Claim Presented					\$260.00	\$0.00
Total						\$690.00

Please charge Deposit Account **09-0464** for the **Total** set forth above. The Commissioner is authorized to charge payment of any additional filing fees required under 37 CFR §1.16 and any patent application processing fees under 37 CFR §1.17 or to credit any overpayment to the identified account. A duplicate copy of this sheet is enclosed.

Express Mail Certificate

Express Mail Label No: EK506756349US

Date: April 7, 2000

I hereby certify that I am depositing the papers identified above with the U.S. Postal Service "Express Mail Post Office to Address" service on the above date, addressed to the Commissioner of Patents and Trademarks, Washington, DC 20231

Karen Orzechowski
Karen Orzechowski

BY: *Joselyn G. Cockburn*

Joselyn G. Cockburn

Attorney of Record Reg. No. 27,069

Date: April 7, 2000

IBM Corporation 972/B656

Intellectual Property Law

PO Box 12195

Res. Tri. Park, NC 27709

Telephone: 919-543- 9036 FAX 919-543-3634

METHOD AND SYSTEM FOR PROVIDING REMOTE PROCEDURE CALLS IN A MULTIPROCESSING SYSTEM

FIELD OF THE INVENTION

The present invention relates to multiprocessing systems and more particularly to a method and system for providing remote procedure calls in such systems.

BACKGROUND OF THE INVENTION

The multiprocessor, parallel execution model used in a general network processor design demands an effective, simple implementation of remote procedure calls. Remote calls may be made from a general purpose processor to a network processor pico-engine, or generically, between any processors within the system. The execution of remote procedure calls enables flexible optimized distribution of both function and workload among multiple processors. Two methods are currently utilized to provide this optimized distribution.

One solution is to pass the entire instruction address within the control message. This successfully effects the remote procedure call, but how does the originator of the message know the instruction address of the call? If the processors use independent code loads, what if the entry point for the remote procedure call execution changes from code load to code load? Some external processing is required to analyze code images to verify/determine entry points and assist in resolving entry point to address at remote procedure call issue time.

An alternative solution is to create a jump table for the processor executing the remote procedure call. A remote procedure call is requested by issuing a request with the appropriate index into the jump table. Thus, the only consistency required between remote procedure call network processor requester and remote procedure call network processor receiver is that the

table index remain consistent in both processes. This approach works well, but suffers from requiring a pre-allocated jump table. In the multiprocessor environment memory storage is at a premium. Thus it is desirable to minimize the amount of dedicated storage in the memory.

Accordingly, what is needed is a system and method which overcomes the above-identified problem. The present invention addresses such a need.

SUMMARY OF THE INVENTION

A method and system for providing remote procedure calls in a multiprocessing system is disclosed. The multiprocessing system includes a general purpose processor and a plurality of network processors. Each of the plurality of network processors includes a memory. The method and system comprises accessing a reserved address in at least one of the network processors and initiating a software action by a first portion of the reserved address. The method and system further includes pointing to an address within the memory of the at least one network processor to be processed based upon data in a second portion of the reserved address, wherein the data at the address is processed.

A system and method in accordance with the present invention provides an indirect software jump in a microprocessing system through providing a reserved address in memory of each of the reserved address includes two portions. A first portion of the address triggers the software event requested by the general purpose processor (for example) and a second portion of the reserved address is utilized to process the data that was loaded at that address in the processor. The indirect software jump allows a general purpose processor to execute software on a network processor indirectly for custom application services or debug operations.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a conventional multiprocessing system.

Figure 2 illustrates such a memory architecture.

Figure 3 illustrates the use of a jump table resource on the network processor

Figure 4 is a block diagram illustrating a jump table.

Figure 5 is a diagram of a reserved address within a network processor utilized to provide the indirect software jump.

Figure 6 is a flow chart for providing an indirect software jump in a multi-processing system.

DETAILED DESCRIPTION

The present invention relates to multiprocessing systems and more particularly to a method and system for providing remote procedure calls in such systems. The following description is presented to enable one of ordinary skill in the art to make and use the invention and is provided in the context of a patent application and its requirements. Various modifications to the preferred embodiment and the generic principles and features described herein will be readily apparent to those skilled in the art. Thus, the present invention is not intended to be limited to the embodiment shown but is to be accorded the widest scope consistent with the principles and features described herein.

Figure 1 illustrates a conventional multiprocessing system 10. The conventional multiprocessing system 10 includes a general purpose processor 12 which includes an instruction set. The general purpose processor 12 controls a plurality of network processors 14, 16 and 18 via a switch fabric 20. Each of the network processors 14 through 18 are

independent central processing units. The network processors 14, 16, 18 are special processors that provide protocol forwarding and are highly tuned for network programming. The switch fabric 20 allows for combinations of processors 12-18 to communicate with each other.

In each of the processors 12-18, there is dedicated memory which holds data and code.

5 Figure 2 illustrates such a memory architecture 100. As is seen data 102 comprises upper portion of the memory structure and code 104 for executing the data occupies the lower portion. The network processors 14-18 (Figure 1) could be the same version of a processor (such x86, Apple, Pentium or RISC or other processor architecture). However, it is more likely that the network processors are different versions of a particular processor family.

10 Accordingly, the memory architecture may be different for each processor so a software operation that is called by one processor to be preformed b other processors may have to be initiated from a different part of the memory architecture from processor to processor. In addition, memory is a very precious resource, particularly for the network processors. The network processors are highly tuned for protocol handling and therefore do not have a large amount of memory to dedicate to a particular function. As the number of network processors grow, the number of commands that can be sent to a processor increases.

15 The multiprocessor, parallel execution model used in a network processor design demands an effective, simple implementation of remote procedure calls. Remote calls may be made from the general purpose processor 12 to one or more network processor 14-18, or generically, between any of the processors 12-18 within the system 10. The execution of remote procedure calls enables flexible optimized distribution of both function and workload among multiple processors. To provide for this flexibility, one of two conventional systems are utilized.

In a first system, the entire instruction address is passed within a control message. This successfully provides for the remote procedure call, but the originator of the message does not know the instruction address of the call. When the processors use different independent code loads, sometimes the entry point for the remote procedure call execution changes from code load on one processor to a different code load on a second processor. Some external processing is required to analyze code images to verify/determine entry points and assist in resolving entry point to address at remote procedure call issue time for each of the processor.

An alternative system is to create a jump table for the processor executing the remote procedure call. Figure 3 illustrates the use of a jump table resource on a network processor 14'. A remote procedure call is requested by issuing a request with the appropriate index into the jump table 200 via the general purpose processor 12'. Thus, the only consistency required between the remote procedure call requester and remote procedure call processor is that the table index remain consistent in both processors. This approach works well, but suffers from requiring a pre-allocated jump table 200.

To explain the problem with a pre-allocated jump table 200, refer now to the following discussion in conjunction with the accompanying figures. Figure 4 is a block diagram illustrating a jump table 200. The jump table 200 includes a plurality of entries which each of which includes an offset and an address. The jump table 200 must be accessed by a control message from a processor that indicates a software action. Next the processor must load a base address and add that to the starting address. Thereafter, the software executes from this new address. Accordingly contiguous memory blocks must be allocated in the memory for the jump table and there is always some of the table that is not used and therefore can be utilized for no other purpose.

Accordingly, a significant amount of reserved storage for either maintaining a pre-allocated jump table or a large amount of external processing is required to analyze code images to determine what the starting instruction address is for the software action adding to the latency of the system.

5 In a system and method in accordance with the present invention an indirect software jump is provided through reserving an address in memory where the address includes two portions. A first portion triggers the software event and a second portion of the address is utilized to process the data that was loaded into a register in the processor. An indirect software jump allows a general purpose processor to execute software on a network processor indirectly for custom application services or debug. An example of such a service is performing a custom table insert which inserts a forwarding entry and automatically inserts a pointer to a global counter block specific to that network processor.

To describe the present invention in more detail, refer now to the following discussion in conjunction with the accompanying figures. Figure 5 is a diagram of a reserved address 300 within a network processor utilized to provide the indirect software jump. The reserved address 300 is typically a four to eight byte value dependent upon the length of the instruction. The address includes a first portion 302 for triggering the software action and a second portion 304 for pointing to the address of memory that is to be processed. Each of the reserved addresses 300 within a processor are located in a known area (preferably in the same location for each memory).

Figure 6 is a flow chart for providing an indirect software jump in a multi-processing system. First, a reserved address is accessed by a command, via step 402. Next, a software action is initiated by a first portion of the reserved address, via step 404. Then, the second

portion of the address points to an address that is to be processed within the memory, and wherein the data at that address is processed, via step 406.

By utilizing an indirect software jump, the remote procedure call executor can de-reference the requested operation within the scope of its own load module. External processing is not required to resolve entry points. A preallocated jump table is not precluded, but is not required. Since the full address of the remote procedure call (or function) pointer is provided in the place of an index, the pointers can be arranged in a tabular form or anchored individually throughout the memory space. The only requirement is that the address of the function pointer, or remote procedure call pointer, remain constant between remote procedure call caller and processor. Therefore, allocation of memory is minimized and contiguous memory to execute an indirect software jump is also not required.

Conclusion

A system and method in accordance with the present invention provides an indirect software jump in a microprocessing system through providing a reserved address in memory of each of the reserved address includes two portions. A first portion of the address triggers the software event requested by the general purpose processor (for example) and a second portion of the reserved address is utilized to process the data that was loaded at that address in the processor. The indirect software jump allows a general purpose processor to execute software on a network processor indirectly for custom application services or debug operations.

Although the present invention has been described in accordance with the embodiments shown, one of ordinary skill in the art will readily recognize that there could be

1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2

CLAIMS

What is claimed is:

1 1. A method for providing remote procedure calls in a multiprocessing system, the
2 multiprocessing system including a general purpose processor and a plurality of network
3 processors; each of the plurality of network processors having a memory, the method
4 comprising the steps of:

- 5 (a) accessing a reserved address in at least one of the network processors;
6 (b) initiating a software action by a first portion of the reserved address; and
7 (c) pointing to an address within the memory of the at least one network
8 processor to be processed based upon data in a second portion of the
9 reserved address; wherein the data at the address is processed.

1 2. The method of claim 1 wherein the reserved address comprises one instruction.

1 3. The method of claim 1 wherein each of the network processors include a
2 reserved address.

1 4. The method of claim 1 wherein a location of the reserved address of each
2 network processor is known by the other processors.

1 5. The method of claim 4 wherein the reserved addresses of each network
2 processor is in the same location of memory.

1 6. A system for providing remote procedure calls in a multiprocessing system, the
2 multiprocessing system including a general purpose processor and a plurality of network
3 processors; each of the plurality of network processors having a memory, the system
4 comprising:

5 means for accessing a reserved address in at least one of the network
6 processors;

7 means for initiating a software action by a first portion of the reserved address;
8 and

9 means for pointing to an address within the memory of the at least one network
10 processor to be processed based upon data in a second portion of the reserved address; wherein
11 the data at the address is processed.

1 7. The system of claim 6 wherein the reserved address comprises one instruction.

1 8. The system of claim 6 wherein each of the network processors include a
2 reserved address.

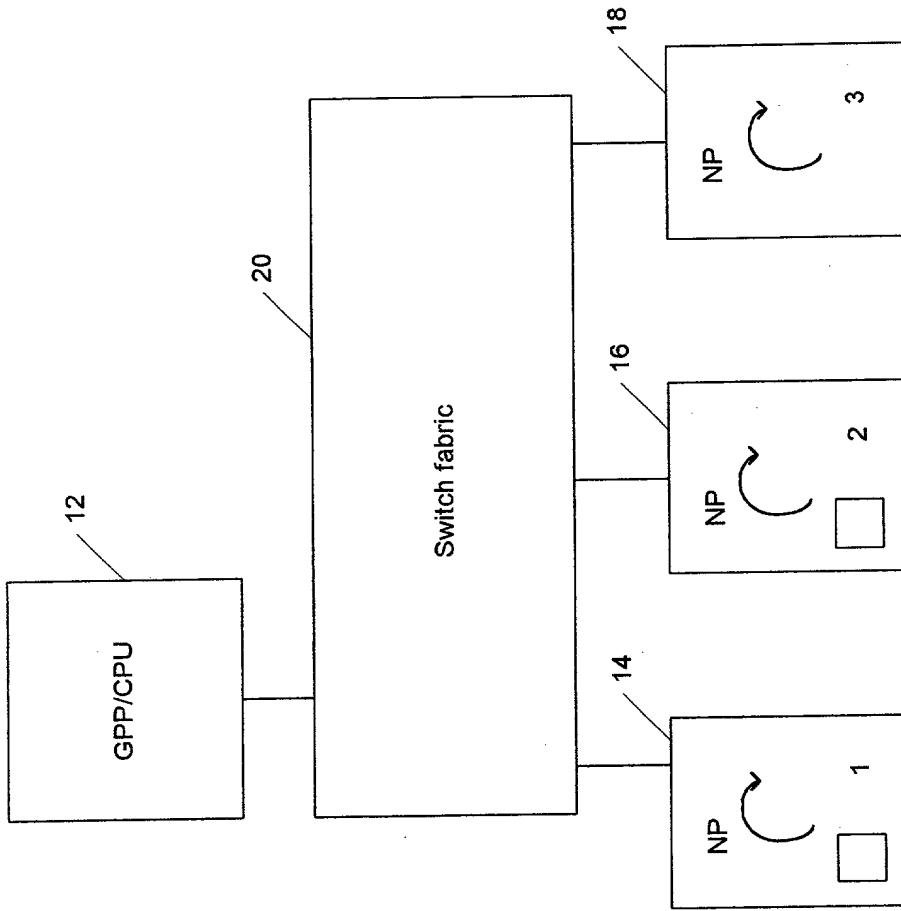
1 9. The system of claim 6 wherein a location of the reserved address of each
2 network processor is known by the other processors.

1 10. The system of claim 9 wherein the reserved addresses of each network
2 processor is in the same location of memory.

ABSTRACT

A method and system for providing remote procedure calls in a multiprocessing system is disclosed. The multiprocessing system includes a general purpose processor and a plurality of network processors. Each of the plurality of network processors includes a memory. The method and system comprises accessing a reserved address in at least one of the network processors and initiating a software action by a first portion of the reserved address. The method and system further includes pointing to an address within the memory of the at least one network processor to be processed based upon data in a second portion of the reserved address, wherein the data at the address is processed. A system and method in accordance with the present invention provides an indirect software jump in a microprocessing system through providing a reserved address in memory of each of the reserved address includes two portions. A first portion of the address triggers the software event requested by the general purpose processor (for example) and a second portion of the reserved address is utilized to process the data that was loaded at that address in the processor. The indirect software jump allows a general purpose processor to execute software on a network processor indirectly for custom application services or debug operations.

002040" 6354560



10

Figure 1

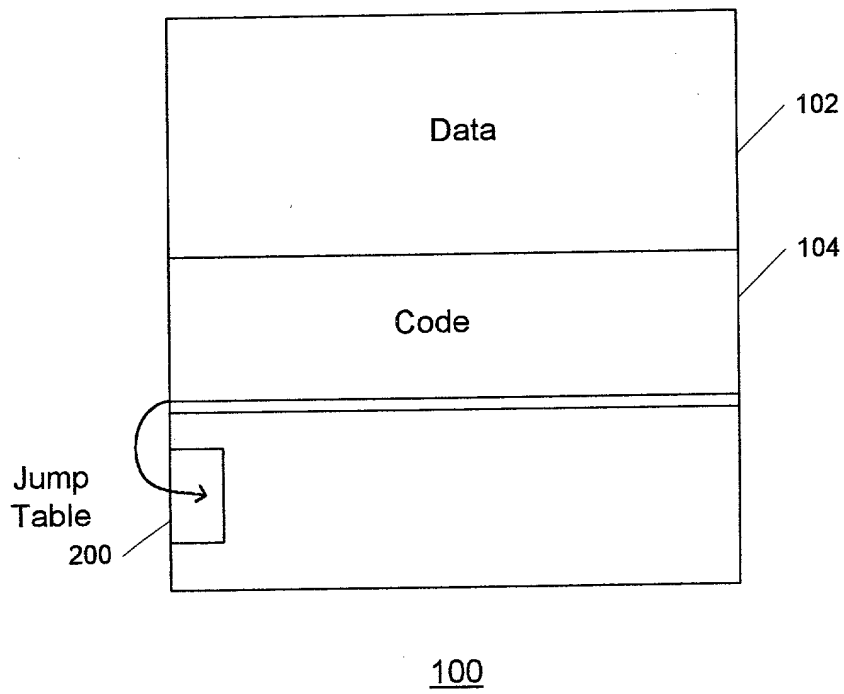


Figure 2

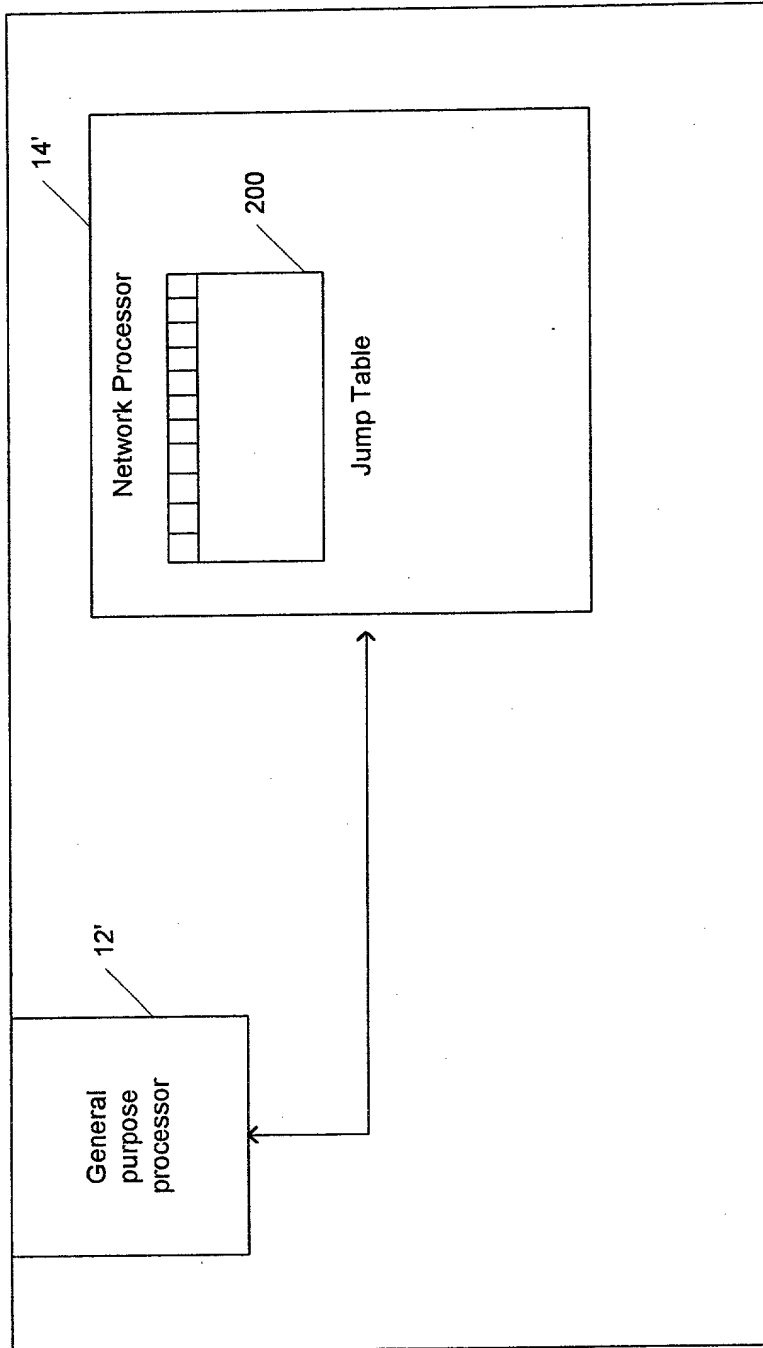


Figure 3

index 0	address 0
index 1	address 1
index 2	address 2
index 3	address 3
index 4	address 4
index 5	address 5
index n	address n



Figure 5

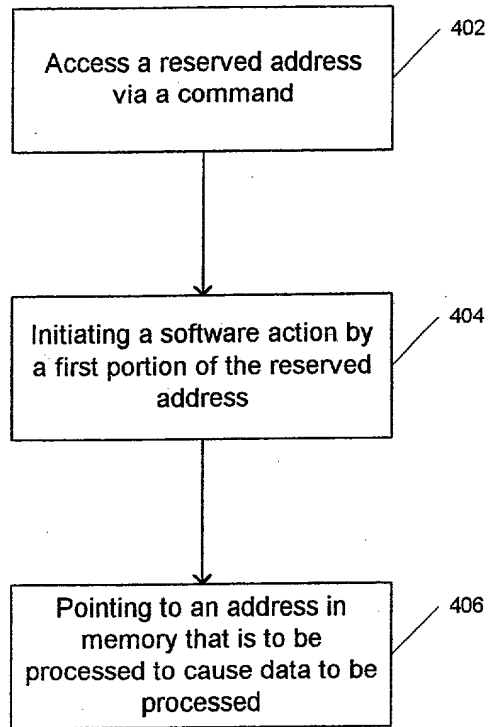


Figure 6

002040" 6354560

**DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name; I believe I am an original, first and joint inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**Method and System for Providing Remote Procedure Calls in a Multiprocessing
System**

the specification of which is identified by the attorney (IBM) Docket Number appearing above.

I hereby state that I have reviewed and understand the contents of the above- identified specification, including the claims.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s)

<u>Number</u>	<u>Country</u>	<u>Day/Month/Year</u>	<u>Priority Claimed</u>
---------------	----------------	-----------------------	-------------------------

I hereby claim the benefit (a) under Title 35, United States Code, §119(e) of any U.S. application listed below and identified as a provisional application or (b) under Title 35, United States Code, §120 of any U.S. application listed below and not identified as a provisional application, and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior U.S. application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application

Prior U.S. Applications

<u>Serial No.</u>	<u>Filing Date</u>	<u>Status</u>
-------------------	--------------------	---------------

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith: Daniel E. McConnell, Reg. No. 20,360; Kenneth A. Seaman, Reg. No. 28,113; Joscelyn G. Cockburn, Reg. No. 27,069; Gerald R. Woods, Reg. No. 24,144; John D. Flynn, Reg. No. 35,137; Horace St. Julian, Reg. No. 30,329; Joseph C. Redmond, Jr., Reg. No. 18,753; John E. Hoel, Reg. No. 26,279; Christopher A. Hughes, Reg. No. 26,914; and Edward A. Pennington, Reg. No. 32,588.

Send all correspondence to: Joscelyn G. Cockburn, IBM Corporation 972/B656; PO Box 12195; Research Triangle Park, NC 27709.

First Inventor: Richard John Blasiak

Signature: Richard John Blasiak 4/6/2000
Date

Residence: 510 Bayberry Drive, Chapel Hill, NC 27514

Citizenship: USA

Post Office Address: same as above

Second Inventor: David Wayne Cosby

Signature: David Wayne Cosby 4/6/2000
Date

Residence: 12209 Queensbridge Court, Raleigh, NC 27613

Citizenship: USA

Post Office Address: same as above

Third Inventor: Anthony Matteo Gallo

Signature: Anthony Matteo Gallo 4/5/2000
Date

Residence: 3308 Corsham Drive, Apex, NC 27502

Citizenship: USA

Post Office Address: same as above

Fourth Inventor: Frank Andrew Smoak

Signature:

Frank Andrew Smoak

5 April 2000
Date

Residence: 402 New Rail Drive, Cary, NC 27513

Citizenship: USA

Post Office Address: same as above

004000683450